



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/539,762	06/20/2005	Eun-Jeong Choi	HI-0189	5901
34610 7590 06/10/2011 KED & ASSOCIATES, LLP P.O. Box 8638 Reston, VA 20195				
EXAMINER CHEMPAKASERIL, ANN J				
ART UNIT		PAPER NUMBER		
2166				
MAIL DATE		DELIVERY MODE		
06/10/2011		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/539,762

**Applicant(s)**

CHOI, EUN-JEONG

**Examiner**

ANN CHEMAKASERIL

**Art Unit**

2166

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 28 October 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 30-33 and 35-62 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 30-33, 35-62 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-942)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

**DETAILED ACTION**

1. Claims 30-33, 35-62 are pending in the action.

***Response to Arguments***

2. Applicant's arguments filed 10/28/2010 have been fully considered but they are not persuasive for the following reasons.

In response to Applicants arguments that Sahota does not disclose a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents, and generates a tree-based object on the basis of graphic user interface (GUI) of the terminal, the tree-based object corresponding to the application program, examiner disagrees. Sahota discloses XML files or documents that are created can be used by content harvest and conversion platform 130 and syndication server 110 and can be defined for specific types of applications and audiences using document type definitions (DTDs). DTD defines the way an XML document should be constructed and generating a tree –based object [0095] on the basis of graphic user interface (GUI) of the terminal [Figure 9b]

***Claim Rejections - 35 USC § 112***

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 30-33, 38-54, and 57-62 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claims 30, 38, 44, 48, 49, 57, 58, 62 recites the tree-based object corresponding to the application program. The specification in the published application states " An XML verifier 322 and a GUI-based object generator 323 of the syntax parser 320 parse the syntax through the contents model 321 on the basis of DTD of the document, forms a tree-based object on the basis of GUI of the terminal 210 and provides the tree-based object to a rendering editor." [0114] However support was not found for the claimed the tree-based object corresponding to the application program.

Claim 30, 38, 44, 48, 49, 57, 58, 62 recites "a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents model..." However the specification states "The syntax parser 320 parses a contents model 321 on the basis of DTD of each document, parses each syntax on the basis of the result of the parsing the contents model 321, and generates a tree-based object on the basis of GUI of the terminal to provide the tree-based object as the rendering data." [0059] Support has not been found for a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents model.

Dependant claims are also rejected for inheriting the deficiencies of the base claims.

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 30, 35, 37, 38, 44, 48, 49, 55, 57, 58 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 30 recites the limitation "the basis of markup and non-markup" in line 9. There is insufficient antecedent basis for this limitation in the claim.

Claim 30 recites the limitation "a corresponding parser" It is unclear to the examiner whether the limitation relates to "an integral parser" or "a word parser" or is a different element and if different how they relate to each other.

Claim 30 recites "a different token according to whether it is a markup or a non-markup". It is unclear to the examiner whether the limitation is related "each different token" in line 11.

Claim 30 recites "a markup or a non-markup" It is unclear whether the elements relate to "a predetermined markup language" in line 3 or "a markup language" in line 4 or " markup and non-markup" in line 10

Claim 30 recites the limitation "the basis of the tokens" in line 14. There is insufficient antecedent basis for this limitation in the claim.

Claim 30 recites "the tokens" in line 15. There is insufficient antecedent basis for this limitation in the claim. It is unclear to the examiner whether "the tokens" is the same

as or different from "a token" in line 9, "each different token" in line 11, "a different token" in line 12 or another element and if different, how they relate to each other.

Claim 30 recites the limitation "the basis of a graphic user interface" in line 17. There is insufficient antecedent basis for this limitation in the claim.

Claim 30 recites the limitation "a specific markup language by the GUI of the handheld terminal" in line 19. It is unclear whether a specific markup language is related to "a markup language" in line 4.

Claim 30 recites the limitation "the parsed markup web-document" in line 20. There is insufficient antecedent basis for this limitation in the claim.

Claim 30 recites "the tokens" in lines 21-22. It is unclear to the examiner whether "the tokens" is the same as or different from "a token" in line 9, "each different token" in line 11, "a different token" in line 12 or another element and if different, how they relate to each other.

Claim 30 recites the limitation "a document" in line 22. It is unclear whether a document relates to "a web document", "a kind of document", "the parsed markup web-document" or a different element and if different how they relate to each other.

A similar rationale of 112 rejections used to reject claim 30 can also be applied to claim 38, 44, 50, 54.

Claim 31 recites "a token" in line 4. It is unclear to the examiner whether "a tokens" is the same as or different from elements in Claim 30, "a token" in line 9, "each different token" in line 11, "a different token" in line 12 or another element and if different, how they relate to each other.

Claim 31 recites "a token" in line 5. It is unclear to the examiner whether "a tokens" is the same as or different from elements in Claim 30, "a token" in line 9, "each different token" in line 11, "a different token" in line 12 or another element and if different, how they relate to each other.

Claim 31 recites "a token" in line 6. It is unclear to the examiner whether "a tokens" is the same as or different from elements in Claim 30, "a token" in line 9, "each different token" in line 11, "a different token" in line 12 or elements in Claim 31, "a token" in lines 4 or 5, or another element and if different, how they relate to each other.

A similar rationale of 112 rejections used to reject claim 31 can also be applied to claim 40, 45, 51.

Claim 32 recites the limitation "a corresponding web-document" in line 3. It is unclear whether it relates to elements in Claim 30, "a web document" in line 1, "a kind of document" in line 10, "the parsed markup web-document" in line 20, "a document" in line 22 or a different element and if different how they relate to each other.

Claim 32 recites the limitation "the basis of the token" in line 4. There is insufficient antecedent basis for this limitation in the claim.

Claim 32 recites the limitation "the analyzed markup web-document" in line 5. There is insufficient antecedent basis for this limitation in the claim

A similar rationale of 112 rejections used to reject claim 32 can also be applied to claim 43, 47, 53.

Claim 33 recites the limitation "the parsing system" in lines 1-2. There is insufficient antecedent basis for this limitation in the claim. It is unclear whether the

parsing system relates to elements in Claim 30, an integral parser, a word parser, or a syntax parser, or a different element.

Claim 33 recites the limitation "a web-document" in line 2. It is unclear whether it relates to elements in Claim 30, "a web document" in line 1, "a kind of document" in line 10, "the parsed markup web-document" in line 20, "a document" in line 22 or a different element and if different how they relate to each other.

Claim 33 recites the limitation "the basis of any one of SGML and XML" in line 4. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "the result" in line 4. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "the case" in line 17. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "the case" in line 21. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "the basis of an element" in line 25. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "an element" in line 25. It is unclear if an element is the same or different from "an element" in lines 11-12 and if different how they relate to each other.

Claim 35 recites the limitation "the undefined token" in line 26. There is insufficient antecedent basis for this limitation in the claim.



Claim 35 recites the limitation "the tokens" in line 27. There is insufficient antecedent basis for this limitation in the claim.

Claim 35 recites the limitation "a document" in line 27. It is unclear if a document is the same or different from "a web document" in line 1 and if different how they relate to each other.

Claim 35 recites the limitation "the basis of markup and non-markup" in line 25. There is insufficient antecedent basis for this limitation in the claim

A similar rationale of 112 rejections used to reject claim 35 can also be applied to claim 37, 55.

Claim 39 recites "all documents" in line 4. It is unclear to the examiner whether all documents includes a web document, a parsed markup web-document, a document as recited in claim 38.

Claim 41 recites "the supplied web-document" in line 3. There is insufficient antecedent basis for this limitation in the claim.

Claim 41 recites "a token" in line 3. It is unclear if a token is the same or different from a token in claim 38, and if different how they relate to each other.

Claim 41 recites the limitation "the basis of the token table" in lines 3-4. There is insufficient antecedent basis for this limitation in the claim

A similar rationale of 112 rejections used to reject claim 41 can also be applied to claim 46, 52.

Claim 42 recites "all documents" in lines 2-3. It is unclear to the examiner whether all documents includes a web document, a parsed markup web-document, a document as recited in claim 38.

Claim 48 recites the limitation "the document" in line 9. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites the limitation "the basis of the tokens" in line 9. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites the limitation "the tokens" in line 9. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites the limitation "the basis of a Graphic User Interface" in line 9. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites the limitation "a specific markup language by the GUI of the handheld terminal" in line 12. It is unclear whether a specific markup language is related to "a markup language" in line 4.

Claim 48 recites the limitation "the parsed markup document" in line 9. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites the limitation "tokens of the web-document" in line 17. It is unclear with tokens are related to "the tokens" in line 9.

Claim 48 recites "a markup language" in line 18. It is unclear to the examiner if "a markup language" is the same as "a markup language in line 4 " or a predetermined markup language" in line 3, or a different element.

Claim 48 recites the limitation "the tokens" in line 21. There is insufficient antecedent basis for this limitation in the claim

Claim 48 recites "a document" in line 21. It is unclear to the examiner whether all documents includes a web document, a parsed markup web-document, or a document or a different element.

Claim 48 recites the limitation "the basis of markup and non-markup" in line 9. There is insufficient antecedent basis for this limitation in the claim

A similar rationale of 112 rejections used to reject claim 41 can also be applied to claim 49, 57, 58.

Claim 60 recites the limitation "the terminal" in line 2. There is insufficient antecedent basis for this limitation in the claim

Claim 60 recites the limitation "the parsed character data" in line 2. There is insufficient antecedent basis for this limitation in the claim

Claim 61 recites the limitation "the basis of the tokens" in line 11-122. There is insufficient antecedent basis for this limitation in the claim

Claim 60 recites the limitation "the tokens" in line 12. There is insufficient antecedent basis for this limitation in the claim

Claim 60 recites the limitation "the handheld" in line 13. There is insufficient antecedent basis for this limitation in the claim

Dependant claims are also rejected for inheriting the deficiencies of the base claims.

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 30-33, 38, 39, 40-45, 46-49, 50-54, 57-59, 62 are rejected under 35 U.S.C. 103(a) as being anticipated by U.S. Patent Application Publication 2001/0056460 issued to Sahota et al. (hereinafter "Sahota") and further in view of US PGPub 2004/0054535 issued to Mackie et al. (hereinafter "Mackie")

As per claims 30, 38, 62, Sahota discloses parsing a web-document based on elements, which is provided to an application of a handheld terminal when the system calls the web-document from a sever to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, to provide it to the handheld terminal (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 [0065] based on elements [0104] which is provided as a template to for a mobile device such as a wireless telephone or personal data assistant (PDA) [0040] when access to a web page is requested [0044]), comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal (The syndication processing techniques described herein can be implemented

by a processor or central processing unit (CPU) executing code or instructions stored in a machine-readable medium. [0029]), the integral parser comprising:

a word parser that separates a token on the basis of markup and non-markup by referring to a token table for all markup data necessary for kind of a document to be supported, wherein a same string of the web document has a different token depending on whether it is a markup on a non-markup in contrast to a general programming language. (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 on the basis of markup and non-markup such as selected tags, styles, and content are either replaced or removed from the HTML page [0065], Sahota explains that content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] If pure data can be extracted and the data us separated from the attribute while using html, then there is inherently an identifier for identifying comments and markup language.),

wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax (integral) parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non- markup portion, including parsed character data for a user (data from the content is extracted using the created capture templates. For example, content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and

attributes. [0041] Content engine 202 dynamically composes content, scripts, and media for syndication server 110. Content engine 202 can also generate code or instructions, which are industry standard compliant, to exploit specific platforms and devices. For example, content engine 202 can convert an HTML web page into an XML file. [0051] an HTML web page is parsed. For example, HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213. In one embodiment, selected tags, styles, and content are either replaced or removed from the HTML page. [0065]).

a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents, and generates a tree-based object on the basis of graphic user interface (GUI) of the terminal, the tree-based object corresponding to the application program (XML files or documents that are created can be used by content harvest and conversion platform 130 and syndication server 110 and can be defined for specific types of applications and audiences using document type definitions (DTDs). DTD defines the way an XML document should be constructed and generating a tree – based object [0095] on the basis of graphic user interface (GUI) of the terminal [Figure 9b])

and performs a mapping operation so as to represent a GUI model of a specific markup language by GUI supported by the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal (In particular, rather than having to change the data because of a different representation, an XSL application provides a complete separation of data,

content, and presentation. Specifically, an XSL application is used to map an XML file into another representation. Using XSL is thus comparable to writing a Java program to translate data into, e.g., a PDF or HTML document, but supplies a standard interface to accomplish such a task. [0068] The parsing system can be applied to multiple platforms or formats such as, for example, HTML, portable document format (PDR), Postscript, or other like formats and architectures such as, for example, a personal computer or an electronic portable device. [0034])

Sahota does not explicitly disclose wherein each different token is generated by a corresponding parser.

Mackie teaches the claimed each different token is generated by a corresponding parser (The parser rule knowledge base includes a predetermined set of parser rules in which each parser rule defines a complex constituent according to a predetermined pattern of tokens and/or simplex constituents and/or complex constituents. Thus, the complex constituent spans a sequence of at least one token in the tokenized text. [0024-0025])

Sahota and Mackie are analogous art because they relate to structured text processing.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to modify the system as taught by Sahota to teach that each different token is generated by a corresponding parser as taught by Mackie.

Modification would allow an interpretation of the message elements of the corresponding structured text for a useful purpose. [Abstract]

As per claims 31, 40, 45, 51, Sahota teaches the system of claim 30(HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 on the basis of markup and non-markup such as selected tags, styles, and content are either replaced or removed from the HTML page [0065])

Sahota does not explicitly teach, "a comment parser for processing a comment and a space; a markup start parser for recognizing a markup start tag and generating a token; an attribute parser for parsing an attribute and generating a token; and a parsed character data analyzer for analyzing parsed character data and generating a token" as claimed.

Mackie teaches the claimed a comment parser that processes a comment and a space (A predetermined parser rule such as the comment parser processes a word of structured text delimited by whitespace [0024-0025]);

a markup start parser that recognizes a markup start tag and generates a token (A token is generated when the markup starter parser recognizes a start label [0029]);

an attribute parser that parses an attribute and generates a token (an attribute parser that parses an attribute and generates a token [0042-0043]; see example [0045]);

and a parsed character data analyzer that analyzes parsed character data and generates a token, (parsing character data and generating a token [0045]).



Sahota and Mackie are analogous art because they relate to structured text processing.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to modify the system as taught by Sahota to use content parsers and generate a token based on the context as taught by Mackie in order to make a more precise decision regarding action to be taken on token. (Mackie, [Abstract]). Modification would allow an interpretation of the message elements of the corresponding structured text for a useful purpose. [Abstract]

Sahota teaches the claimed wherein the attribute parser is configured to recognize a name of an attribute or to recognize a value of an attribute (The "@<attribute name>" function, e.g., @SRC - Starting from the first <IMG>, retrieves the value for SRC, =http://image.weather.com/pics/blank.gif. The "All" function, starting from the first <FONT>, retrieves one long string containing all elements, attributes and text, as shown below. [0133])

As per claim 32, 43, 47 and 53, Sahota discloses the syntax parser comprises: an XML verifier that verifies whether a corresponding document is composed suitable for each DTD on the basis of the token generated by the word parser (HTML converter 208 converts existing HTML type content into clean well-formed documents (XHTML) for conversion into XML service specific schemas and data files. An XML schema offers an XML centric means to constrain XML documents. [0059] ML files or documents that are created can be used by content harvest and conversion platform 130 and

syndication server 110 and can be defined for specific types of applications and audiences using document type definitions (DTDs) [0136; Sahota]);

and a terminal GUI-based object generator that matches the analyzed markup web-document and the GUI of the handheld terminal (By generating a standardized data stream from the capture templates, content can be displayed on multiple types of platforms [0025]).

As per claim 33, Sahota discloses the parsing system integrally parses a web-document composed on the basis of any one of SGML and XML related to HTML, XHTML, mHTML, cHTML, WML and HDML (Parsing a web-document composed of XML related to HTML [0026], WML [0006], XHTML [0059])

As per claim 39, Sahota discloses the token table comprises: tokens defined in an XML document (HTML converter 208 creates XML data files 208a based on the conversion rules in the repository and creates XML data files and streams;

keywords defined in DTD for all documents provided to the handheld terminal (Creates XML data files and streams that are used by content converter 204 and content generator 203 subsystems);

a list of elements which can be supported by each terminal (create dynamically content for specific platforms and device frameworks. [0059])

As per claim 41, 46, and 52, Sahota discloses the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-

formedness verifier (HTML converter 208 converts existing HTML type content into clean well-formed documents (XHTML) for conversion into XML service specific schemas and data files. An XML schema offers an XML centric means to constrain XML documents. The conversion logic and process is stored in a content acquisition and conversion rules repository 207a. HTML converter 208 creates XML data files 208a based on the conversion rules in the repository and creates XML data files and streams that are used by content converter 204 and content generator 203 subsystems to create dynamically content for specific platforms and device frameworks. [0059])

As per claim 42, Sahota discloses the contents model means a hierarchy of elements and an attribute list, and is defined in DTD for all documents provided to the handheld terminal. (DTD establishes a set of constraints for an XML file or document. That is, a DTD defines the way an XML document should be constructed. [0136] A hierarchy of elements and attribute list is defined depending on the type of DD; see Table 2.)

As per claim 44, 50, and 54, Sahota discloses parsing web-document based on elements when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 [0065] based on elements [0104] which is provided as a template to for a mobile device such as a wireless telephone or personal data assistant (PDA) [0040] when access to a web page is requested [0044]), comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal (The syndication processing techniques described herein can be implemented by a processor or central processing unit (CPU) executing code or instructions stored in a machine-readable medium. [0029])

a token table comprising tokens defined in an XML document, keywords defined in document type definition (DTD) for documents provided to a handheld terminal, and a list of elements, which can be supported by each handheld terminal (HTML converter 208 creates XML data files 208a based on the conversion rules in the repository and creates XML data files and streams that are used by content converter 204 and content generator 203 subsystems to create dynamically content for specific platforms and device frameworks. [0059];

a word parser for extracting and separating tokens of the web-document supplied to the handheld terminal regardless of kind of a markup language used to compose the web-document by referring to the token table, wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of an attribute (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 on the basis of markup and non-markup such as selected tags, styles, and content are either replaced or removed from the HTML page [0065], Sahota explains that content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] If pure data can be extracted and the data is separated

from the attribute while using html, then there is inherently an identifier for identifying comments and markup language. The tree structure is shown along the web page. By viewing the tree structure, attributes of the HTML tree structure can be filtered out. [0095] also see web page content capturing [0104-0135]);

a contents model determined by DTDs for the documents provided to the handheld terminal that includes a hierarchy of elements and an attribute list (DTD establishes a set of constraints for an XML file or document. That is, a DTD defines the way an XML document should be constructed. [0136] A hierarchy of elements and attribute list is defined depending on the type of DD; see Table 2.);

a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and the contents model, generates a tree-based object on the basis of a graphical User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal. (XML files or documents that are created can be used by content harvest and conversion platform 130 and syndication server 110 and can be defined for specific types of applications and audiences using document type definitions (DTDs). DTD defines the way an XML document should be constructed and generating a tree –based object [0095] on the basis of graphic user interface (GUI) of the terminal [Figure 9b] In particular, rather than having to change the data because of a different representation, an XSL application

provides a complete separation of data, content, and presentation. Specifically, an XSL application is used to map an XML file into another representation. Using XSL is thus comparable to writing a Java program to translate data into, e.g., a PDF or HTML document, but supplies a standard interface to accomplish such a task. [0068])

wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax (integral) parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non- markup portion, including parsed character data for a user (data from the content is extracted using the created capture templates. For example, content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] Content engine 202 dynamically composes content, scripts, and media for syndication server 110. Content engine 202 can also generate code or instructions, which are industry standard compliant, to exploit specific platforms and devices. For example, content engine 202 can convert an HTML web page into an XML file. [0051] an HTML web page is parsed. For example, HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213. In one embodiment, selected tags, styles, and content are either replaced or removed from the HTML page. [0065])

As per claims 48, 49, 57, and 58, Sahota discloses, a handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-

document from a sever to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, to provide it to the handheld terminal (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 [0065] based on elements [0104] which is provided as a template to for a mobile device such as a wireless telephone or personal data assistant (PDA) [0040] when access to a web page is requested [0044]), comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal parses the web-document (The syndication processing techniques described herein can be implemented by a processor or central processing unit (CPU) executing code or instructions stored in a machine-readable medium. [0029]),

wherein the integral parser includes a syntax parser that parses a plurality of syntax of the document on the basis of the tokens and a contents model, generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by GUI supported by the handheld terminal regardless of the specific markup language and thereby matches the parsed markup web-document to the GUI of the handheld terminal (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213. In one embodiment, selected tags, styles, and content are either replaced or removed from the HTML page. Such a modification can be displayed in a

browser to see the changes. [0065] In particular, rather than having to change the data because of a different representation, an XSL application provides a complete separation of data, content, and presentation. Specifically, an XSL application is used to map an XML file into another representation. Using XSL is thus comparable to writing a Java program to translate data into, e.g., a PDF or HTML document, but supplies a standard interface to accomplish such a task. [0068]);

a memory that stores information parsed by the integral parser (Furthermore, the following embodiments describe simple to use application tools for acquiring content and for creating templates to transform the content. The templates can be stored in a file or a database for later, which allows content to be edited such that other types of content can be added to provide new "look and feel" content. [0027]); and

an application program using information extracted from the integral parser, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 on the basis of markup and non-markup such as selected tags, styles, and content are either replaced or removed from the HTML page [0065], Sahota explains that content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] If pure data can be extracted and the



data is separated from the attribute while using HTML, then there is inherently an identifier for identifying comments and markup language. The tree structure is shown along the web page. By viewing the tree structure, attributes of the HTML tree structure can be filtered out. [0095] also see web page content capturing [0104-0135])

wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax (integral) parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user (data from the content is extracted using the created capture templates. For example, content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] Content engine 202 dynamically composes content, scripts, and media for syndication server 110. Content engine 202 can also generate code or instructions, which are industry standard compliant, to exploit specific platforms and devices. For example, content engine 202 can convert an HTML web page into an XML file. [0051] an HTML web page is parsed. For example, HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213. In one embodiment, selected tags, styles, and content are either replaced or removed from the HTML page. [0065])

As per claim 59, Sahota teaches the claimed wherein the attribute parser includes a first attribute parser configured to recognize a name of an attribute and a

second attribute parser configured to recognize a value of the attribute (The "@<attribute name>" function, e.g., @SRC - Starting from the first <IMG>, retrieves the value for SRC , =http://image.weather.com/pics/blank.gif. The "All" function, starting from the first <FONT>, retrieves one long string containing all elements, attributes and text, as shown below. [0133])

8. Claims 35- 37, 55-56, 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sahota and further in view of Chadha et al. (US PGPub 2003/0184552; hereinafter "Chadha").

As per claims 35, 37 and 55, Sahota discloses parsing a web-document called by a handheld terminal from of a web-server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 [0065] based on elements [0104] which is provided as a template to for a mobile device such as a wireless telephone or personal data assistant (PDA)An HTML web page is parsed. A user can access a particular web page on web server. The content from the web page is then parsed [0044]), the method comprising

outputting information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal (The syndication processing techniques described herein can be implemented by a

processor or central processing unit (CPU) executing code or instructions stored in a machine-readable medium. [0029])

reading a token from the web-document and parsing the token using the output information (reading a token from the webpage and parsed for tokens such as tags [0036]);

if the token is not a defined start tag or if the token is a comment or a space as result of the reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token (A token is generated when the markup starter parser recognizes a start label. A predetermined parser rule such as the comment parser processes a word of structured text delimited by whitespace [0024-0025; Mackie]);

parsing the attribute of the element from the token, storing Graphic User Interface (GUI)-related information of the element, and parsing contents of the element (content harvest and conversion platform 130 can be used to extract pure data from the web page such as, for example, the HTML tags and attributes. [0041; Sahota]);

Sahota teaches the claimed as the result of the parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears (HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213 on the basis of markup and non-markup such as selected tags, styles, and content are either replaced or removed from the HTML page [0065], Sahota explains that content harvest and conversion platform 130 can be used to

extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] If pure data can be extracted and the data is separated from the attribute while using HTML, then there is inherently an identifier for identifying comments and markup language. The tree structure is shown along the web page. By viewing the tree structure, attributes of the HTML tree structure can be filtered out. [0095] also see web page content capturing [0104-0135]);

in the case that the contents of the element are parsed character data, performing a mapping operation as to represent a GUI model of a specific markup language by a GUI (Graphic User Interface) of the handheld terminal regardless of the specific markup language, and thereby matching the parsed character data to the GUI of the handheld terminal (In particular, rather than having to change the data because of a different representation, an XSL application provides a complete separation of data, content, and presentation. Specifically, an XSL application is used to map an XML file into another representation. Using XSL is thus comparable to writing a Java program to translate data into, e.g., a PDF or HTML document, but supplies a standard interface to accomplish such a task. [0068] The parsing system can be applied to multiple platforms or formats such as, for example, HTML, portable document format (PDF), Postscript, or other like formats and architectures such as, for example, a personal computer or an electronic portable device. [0034])

wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax (integral) parser ignores only a markup portion of the element that is not

supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non- markup portion, including parsed character data for a user (data from the content is extracted using the created capture templates. For example, content harvest and conversion platform 130 can be used to extract pure data can be extracted from the web page such as, for example, the HTML tags and attributes. [0041] Content engine 202 dynamically composes content, scripts, and media for syndication server 110. Content engine 202 can also generate code or instructions, which are industry standard compliant, to exploit specific platforms and devices. For example, content engine 202 can convert an HTML web page into an XML file. [0051] an HTML web page is parsed. For example, HTML parser engine 217 can parse an HTML web page from Internet content and web sites 213. In one embodiment, selected tags, styles, and content are either replaced or removed from the HTML page. [0065])

Sahota does not explicitly disclose in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears, terminating, and if the end tag corresponding to the defined start tag does not appear, ignoring and returning, wherein the element supported by a terminal for the called web-document is separated from the document by at least one of defining a token table on the basis of an element supported by the terminal and making the undefined token an UNKNOWN token, or ignoring the undefined tag

Chadha teaches the claimed in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears,

terminating, and if the end tag corresponding to the defined start tag does not appear, ignoring and returning, wherein the element supported by a terminal for the called web-document is separated from the document by at least one of defining a token table on the basis of an element supported by the terminal and making the undefined token an UNKNOWN token, or ignoring the undefined tag (The process starts by parsing the next markup language tag in the markup file at step 402. A check is made to determine if there is a tag left at step 416. If there is a tag left, a check is made to determine if the tag is a text-based tag at step 404. If the tag is a text-based tag, a text entry is added in the Object Table and the process returns to step 402 to get the next tag. If the tag is not a text-based tag, a check is made to determine if the tag is for a Graphical User Interface (GUI) based object at step 408. If it for a GUI-based object, then an entry representing the GUI is added into the Object Table at step 410 and the process returns to 402 to get the next tag. If the tag is not a GUI-based tag, a check is made to determine if the tag is a geometry-based tag 412. If the tag is a geometry-based tag, a corresponding entry is made in the Object Table at step 414 and the process returns to step 402 to check if there is any tag left. If the tag is not a geometry-based tag (and therefore not a text-based, GUI-based or geometry-based tag), the tag is ignored at step 418 and the process is returned to 402 to get the next tag. The process ends when there is no markup language tag left to process. [0035]).

Sahota and Chadha are analogous art because they relate to structured text processing.

It would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings Sahota, Mackie, and Chadha to read parsed character data (paragraph [0035]) till the end tag appears. Modification would process the object entries of each of the object types to generate display data corresponding to the object entries.

As per claims 36 and 56, Sahota discloses wherein the parsing comprises the steps of:

if the read token does not include a defined start tag, reading the data continuously until the end tag appears, and if the end tag corresponding to the defined start tag does not appear, thereby ignoring the token (If the tag is not a geometry-based tag (and therefore not a text-based, GUI-based or geometry-based tag), the tag is ignored at step 418 and the process is returned to 402 to get the next tag. [0035]);

reading a new token (The process starts by parsing the next markup language tag in the markup file at step 402. A check is made to determine if there is a tag left at step 416. If there is a tag left, a check is made to determine if the tag is a text-based tag at step 404. The process ends when there is no markup language tag left to process. [0035])

As per claim 60, Sahota does not explicitly teach, "in the case of <p align="center">Hello world!</p>, the terminal that does not support p element ignores markup data between "<" and ">" and browses the parsed character data "Hello world!" for the user" as claimed.

Chadha teaches the claimed in the case of `<p align="center">Hello world!</p>`, the terminal that does not support p element ignores markup data between "<" and ">" and browses the parsed character data "Hello world!" for the user (If the tag is not a geometry-based tag (and therefore not a text-based, GUI-based or geometry-based tag), the tag is ignored at step 418 and the process is returned to 402 to get the next tag. The process ends when there is no markup language tag left to process. [0035])

Sahota and Chadha are analogous art because they relate to structured text processing.

It would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings Sahota, Mackie, and Chadha to parsing text/character data and ignore unsupported/unrecognized HTML tag. Modification would process the object entries of each of the object types to generate display data corresponding to the object entries.

9. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sahota and further in view of Web Design Group (HTML comments)

As per claim 61, Sahota discloses wherein the different tokens are as follows: `<html>` represents an element type, `<p> html</p>` represents parsed character data, `<!--html-->` represents a comment (Figure 9B shows the different tokens, such as element type, and parsed character data);

Sahota does not explicitly disclose the comment feature.



Web design group teaches the claimed `<!--html-->` represents a comment (A *comment declaration* starts with `<!--`, followed by zero or more comments, followed by `>`. A *comment* starts and ends with `--`, and does not contain any occurrence of `--`. )

Web design group and Sahota are analogous art because they are in the same field of endeavor of representing the different tokens

It would have been obvious to use the feature taught by Web Design group and include it as a token for parsing. Modification provides the following simple rule to compose valid and accepted comments.

***Contact Information***

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ann J. Chempakaseril whose telephone number is 571-272-9767. The examiner can normally be reached on Monday through Thursday, 9-4.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571) 272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ann J Chempakaseril/  
Examiner, Art Unit 2166

June 7, 2011